<u>REMARKS</u>

Claims 1-3, 5, 7-9, 11-22, 24-27, 29-31, 35, 36, and 39-45 are currently pending in the subject application and are presently under consideration. Claims 1, 8, 27 and 29 have been amended as shown on pages 2-7 of submission.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claim 1 Under 35 U.S.C §112

Claim 1 stands rejected under 35 U.S.C. §112, second paragraph, due to certain informalities. Withdrawal of this rejection is requested in view of amendment to subject claim.

II. Rejection of Claims 1-3, 5, 7, and 27 Under 35 U.S.C. §102(b)

Claims 1-3, 5, 7, and 27 stand rejected under 35 U.S.C. §102(b) as being anticipated by Sherrod (US 4,642,756). Withdrawal of this rejection is requested since Sherrod fails to teach or suggest all aspects of subject claims.

> A single prior art reference anticipates a patent claim only if it ***expressly or inherently describes each and every limitation*** set forth in the patent claim. *Trintec Industries, Inc. v. Top-U.S.A. Corp.,* 295 F.3d 1292, 63 USPQ2d 1597 (Fed. Cir. 2002); *See Verdegaal Bros. v. Union Oil Co. of California,* 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). ***The identical invention must be shown in as complete detail as is contained in the … claim***. *Richardson v. Suzuki Motor Co.,* 868 F.2d 1226, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Applicants' claimed subject matter relates to systems and methods for specifying and executing temporal order events. The loose specification of events gives users the maximum ability to explore, create variations and make adjustments. The users need not specify exact literal times that an event or task is to be executed. Rather the users need only specify information that that they desire. The rest of the information is inferred by the system. To this end, independent claim 1 recites *a constraint component that receives loose temporal constraints associated with a plurality of events,* ***wherein the loose temporal constraints*** *specify information about execution of an event* ***comprising a start time or a stop time and event execution relative***

to other events, *a system information component that receives* ***execution system information comprising one or more of available memory, cache coherency, data throughput or number of processors****, and an order component that determines... a plurality of event orders in accordance with the loose temporal constraints* ***and via a utility based analysis of the executing system information selects an optimal event execution order from the plurality of event execution orders****.* Sherrod neither teaches nor suggests such novel aspects with respect to, *an order component that determines... a plurality of event orders in accordance with the loose temporal constraints and selects an optimal event order based at least in part on execution system information, wherein the event order specifies the execution order of events.*

Sherrod provides method and apparatus for scheduling the execution of a plurality of processing tasks within a computer system. A task scheduler schedules the execution of a plurality of tasks within a computer system. The task scheduler utilizes a combination of externally assigned priorities and internally calculated priorities to optimize the responsiveness of the computer to external interactions; and this reference fails to teach or suggest each aspect of the claimed subject matter.

At page 3 of Final Office Action, it is erroneously asserted that Sherrod substantially teaches *a constraint component that receives loose temporal constraints associated with a plurality of events, wherein the loose temporal constraints specify information about execution of an event comprising a start time or a stop time and event execution relative to other events*, with respect to independent claim 1. The reference (Sherrod) provides for associating two priority values including an internal priority and an external priority with each task. The internal priority is provided from within a task scheduler and the external priority is assigned by the computer operator (*See*, Col. 4, lines 31- 35). Each task is associated with two timers RUNTIME 10' and INTTIME 11'. These timers are reset to zero by the task scheduler. Both timers count elapsed time whenever the task is being executed by CPU and are suspended (not counting time) whenever the CPU is not executing the task. The RUNTIME timer is reset to zero each time the state of the task is changed. The value in the timer RUNTIME is compared to time parameters such as QUAN1, QUAN1A, QUAN1B, *etc.*, which are expressed in the same time units (*See*, Col. 8, lines 16-17). If the value of the RUNTIME timer exceeds QUAN1C, the state is changed to S$HICP (*See*, Col. 9, lines 1-2). If the value of the RUNTIME timer exceeds QUAN1A, the state is changed to S$CPU (*See*, Col. 9, lines 39-40). Hence Sherrod provides for associating

two timers (RUNTIME 10' and INTTIME 11') with each task that count elapsed time whenever the task is being executed by CPU. The elapsed time counted by the timers is compared to time parameters QUAN1, QUAN1A, QUAN1B, *etc.*, which are set by the computer operator and if the value in the timers (RUNTIME 10' and INTTIME 11') exceeds a particular time parameter (QUAN1, QUAN1A, QUAN1B, *etc.*), the state of the task is changed to a state having lower internal priority and the execution of that task is suspended. More particularly, Sherrod provides for setting time parameters (QUAN1, QUAN1A, QUAN1B, *etc.*) by a computer operator for executing a task for a limited time before suspending the execution of the task and executing another task. It is respectfully submitted that the time parameters set by the computer operator (QUAN1, QUAN1A, QUAN1B, *etc.*) and timers (RUNTIME 10' and INTTIME 11') *are not loose temporal constraints since the execution order of tasks is fully scheduled according to external priority assigned by the computer operator and the execution of a task is suspended depending upon parameters set by the computer operator* (*i.e.* if the value of the RUNTIME timer exceeds QUAN1A, the state is changed to S$CPU) before starting execution of another task. *Hence the whole information about the execution of events is supplied by the computer operator*. Further the two timers (RUNTIME 10' and INTTIME 11') only count *elapsed time* for each task and time parameters (QUAN1, QUAN1A, QUAN1B, *etc.*) only specify duration a task can be executed before being suspended. If the task takes a longer time to execute than the time parameters set by the computer operator (QUAN1, QUAN1A, QUAN1B, *etc.*), the execution of that task is suspended and next task in the queue is executed. However Sherrod does not contemplate *receiving loose temporal constraints associated with a plurality of events, wherein the loose temporal constraints specify information about execution of an event comprising* ***a start time or a stop time*** *and specify event execution relative to other events*. This feature facilitates supporting and executing loosely specified events wherein a user desire to specify only a start time or a stop time or as much information as he desire. Hence a plurality of event orders is possible based upon the loose temporal constraints. An optimal event order is selected based on execution system information such as available memory, cache coherency, number of processors and the like (*See*, Specification, page 22, lines 23-30, Fig. 15).

At page 4 of the Final Office Action, it is erroneously asserted that Sherrod substantially teaches *a system information component that receives execution system information comprising one or more of available memory, cache coherency, data throughput and number of processors*,

with respect to independent claim 1. The reference (Sherrod) provides for associating two priority values including an internal priority and an external priority with each task. The internal priority is provided from within a task scheduler and the external priority is assigned by the computer operator. The internal priority may change from moment to moment as external events related to the task occur; time intervals elapse, or as the task perform input/output operations. The external priorities commonly remain static for the life of the task (*See*, Col. 4, lines 30-44). Hence Sherrod only provides for specifying execution order of events according to external priority set by the computer operator, suspending the execution of an event if the event takes longer time to execute than a fixed time parameter (QUAN1, QUAN1A, QUAN1B, *etc.*), set by the computer operator and executing a task which has higher internal priority at that stage. However Sherrod does not contemplate *a system information component that receives execution system information comprisingone or more of available memory, cache coherency, data throughput or number of processors*. This feature facilitates optimizing the execution order of a series of events. Hence a user can specify a series of constraints loosely. Thus, any number of orderings for events can satisfy the constraints. Systems can choose one of the orderings as the favored ordering. For example, one system could choose to optimize the ordering of events for maximum performance. To accomplish this goal, the system utilizes heuristics or a pool of knowledge about the executing system of which a user who specified the constraints does not know. For example, the system takes into account cache coherency, data throughput, number of number of processors, and available memory.

At page 7 of the Final Office Action, it is erroneously asserted that Sherrod substantially teaches *an order component that determines,* ***a plurality of event orders in accordance with the loose temporal constraints*** *and* ***selects an optimal event order from the plurality of event orders based at least in part on the execution system information****, wherein the event order specifies the execution order of events*, with respect to independent claim 1. The reference (Sherrod) provides for associating two priority values including an internal priority and an external priority with each task. The internal priority is provided from within a task scheduler and the external priority is assigned by the computer operator (*See*, Col. 4, lines 30-34). External task priority values are arranged in three groups *i.e.* fixed-low-priority, fixed-high-priority and interactive-priority based on three parameters, PRI-LOW, PRIHI and PRIMAX, *which are set by the computer operator* (*See*, Table 1). The internal priority for a task is determined by the 'state' of the task. The state

for the fixed-high-priority task is S$RT and is placed at the top of the list and is executed first. The states for the interactive priority tasks are S$TTFN, S$HICP, S$IOFN and S$CPU and executed after the S$RT. The state of the fixed-low-priority task is S$LOW and is executed in last preceding only S$WAIT task which is waiting on some event to occur before execution can proceed (*See*, Table 2). Hence Sherrod only provides for arranging events in a list in descending order of priority depending upon their external priorities, executing and counting the elapsed time (RUNTIME 10' and INTTIME 11') for an event (at the top of the list) when the event is being executed by the CPU, suspending the execution of an event if the event takes longer time to execute than a fixed time parameter (QUAN1, QUAN1A, QUAN1B, *etc.*), set by the computer operator and executing another task which has higher priority. More particularly, Sherrod provides for executing events according to external priority set by the computer operator, suspending the execution of an event if the event takes longer time to execute than a fixed time parameter (QUAN1, QUAN1A, QUAN1B, *etc.*), set by the computer operator and then executing another task which has higher priority. However Sherrod does not contemplate *an order component that determines, via utility-based analysis, **a plurality of event orders** in accordance **with the loose temporal constraints** and **selects an optimal event order from the plurality of event orders based at least in part on the execution system information*** (comprising available memory, cache coherency, data throughput and number of processors)*, wherein the event order specifies the execution order of events.*

In view of at least the foregoing, it is readily apparent that Sherrod fails to teach or suggest all aspects of the claimed subject matter. Accordingly, it is respectfully requested that this rejection of independent claims 1 and 27 (and the claims that depend there from) should be withdrawn.

III. <u>Rejection of Claims 8, 9, 12-19, 22-26, 29-31, 35, 36, 39, 44, and 45 Under 35 U.S.C. §102(b)</u>

Claims 8, 9, 12-19, 22-26, 29-31, 35, 36, 39, 44, and 45 stand rejected under 35 U.S.C. §102(b) as being anticipated by Jerome *et al.* (US 6,323,882). Withdrawal of this rejection is requested since reference (Jerome *et al.*) fails to teach or suggest all aspects of subject claims.

> A single prior art reference anticipates a patent claim only if it ***expressly or inherently describes each and every limitation*** set forth in the patent claim. *Trintec Industries, Inc. v. Top-U.S.A. Corp.,* 295 F.3d 1292, 63 USPQ2d 1597 (Fed. Cir. 2002); *See Verdegaal Bros. v. Union Oil Co. of California,* 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). ***The identical invention must be shown in as complete detail as is contained in the ... claim***. *Richardson v. Suzuki Motor Co.,* 868 F.2d 1226, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Applicants' claimed subject matter relates to systems and methods for specifying and executing temporal order events. To this end, independent claim 8 recites *a display component that provides a plurality of object workspaces, the workspaces are user interfaces including at least one of a past, present and/or future space, the present space is an editable area,* ***the past and future space specify temporal constraints associated with a plurality of events****; and a design component that temporally associates and/or disassociate objects in the editable area and determines an optimal execution order of events based at least in part on the object associations specifying temporal constraints* ***wherein non-associated objects order of execution is determined via utility-based analysis based at least in part on the execution system information comprising available memory, cache coherency, data throughput and number of processors***. Jerome *et al.* neither teaches nor suggests such novel aspects.

Jerome *et al.* provides method and system for creating and using a graphical task scheduler. A Graphical User Interface (GUI) allows a user to graphically build a real time flow task scheduler by providing a "click & drag" functional palette which contains graphical flow chart elements. By creating this graphical flow chart, the user is able to associate a plant layout, or any subpart of the entire plant, to its associated mathematical model. The scheduler controls the running of each sequence based on the scheduling information. Once a plant has been optimized by the graphical task scheduler, the optimized data may be used to control plant operations.

At page 9 of the Final Office Action, it is erroneously asserted that Jerome *et al.* substantially teaches that *non-associated objects order of execution is determined via utility-based analysis, based at least in part on the execution system information comprising available memory, cache coherency, data throughput and number of processors* with respect to independent claim 8. The cited portion of reference (Jerome *et al.*) provides for creating a

sequence by a user. The user opens a sequence process flow diagram (PFD) window. In the PFD window, a unique identifying name of the sequence is displayed in the identification bar. A series of pull down menus and a button bar are provided for the user to interface with the PFD window. *The PFD window includes a sequence display window which provides a graphical display of the current sequence to the user.* The user creates a sequence by using a keyboard, a mouse, or other pointing device such as a trackball or joystick to drop and drag tasks from the task palette into the sequence display window. *To add the task to the current sequence, the user drags the task into the proper location in the sequence display window.* The sequence display window provides a continual visual display to the user of the tasks included in the current sequence (*See*, Col. 5, lines 45-67). Hence Jerome *et al.* provides for creating a sequence by a user and executing the sequence according to schedule specified by the user. More particularly, Jerome *et al.* provides for only executing a scheduled (or associated) sequence. However Jerome *et al.* does not contemplate *a design component that temporally associates and disassociate objects in the editable area and* ***determines an optimal execution order of events*** *based at least in part on the object associations specifying temporal constraints wherein* ***non-associated objects order of execution is determined via utility-based analysis based at least in part on the execution system information comprising available memory, cache coherency, data throughput and number of processors***. This feature facilitates optimizing the execution order of a series of events by performing utility based analysis. Hence a user can specify as much information as they desire. The missing information is subsequently inferred or supplied by other system components. Disassociated objects are represented by providing sufficient space between the objects, so that they are not positioned horizontally next to each other or layered on top of each other. Additionally, operational objects are utilized to further delineated object association and relations. For instance hard edge lines can be used to specify hard start and/or stop times for execution (*See*, Specification, Page 23, lines 5-23 & Fig. 16, 17).

In view of at least the foregoing, it is readily apparent that Jerome *et al.* fails to teach or suggest all aspects of the claimed invention. Accordingly, it is respectfully requested that this rejection of independent claims 8 and 29 (and the claims that depend there from) should be withdrawn.

IV. Rejection of Claims 6 and 43 Under 35 U.S.C. §103(a)

Claims 6 and 43 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Sherrod (US 4,642,756) in view of Howie *et al.* (US 5,093,794). Withdrawal of this rejection is requested for at least the following reasons. Claims 6 and 43 depend from independent claim 1. As stated *supra*, Sherrod do not disclose or suggest every limitation set forth in the subject independent claims. Howie *et al.* fails to make-up for the aforementioned deficiencies of the base combination. Withdrawal of this rejection is therefore respectfully requested.

V. Rejection of Claim 11 Under 35 U.S.C. §103(a)

Claim 11 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Jerome *et al.* (US 6,323,882) in view of Green *et al.* (US 4,646,231). Withdrawal of this rejection is requested for at least the following reasons. Claim 11 depend from independent claim 8. As stated *supra*, Jerome *et al.* do not disclose or suggest every limitation set forth in the subject independent claims. Green *et al.* fails to make-up for the aforementioned deficiencies of the base combination. Withdrawal of this rejection is therefore respectfully requested.

VI. Rejection of Claims 20, 21, and 40-42 Under 35 U.S.C. §103(a)

Claims 20, 21, and 40-42 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Jerome *et al.* (US 6,323,882) in view of Zweben *et al.* (US 5,768,586). Withdrawal of this rejection is requested since Jerome *et al.* and Zweben *et al.* fail to teach or suggest all aspects of subject claims.

> A factfinder should be aware, of course, of the distortion caused by hindsight bias and must be cautious of arguments reliant upon *ex post* reasoning. See *KSR v. Teleflex*, 550 U.S. ___, 127 S. Ct. 1727 (2007) citing Graham v. John Deere Co. of Kansas City, 383 U. S. 1, 36 (warning against a "temptation to read into the prior art the teachings of the invention in issue" and instructing courts to "'guard against slipping into the use of hindsight'" (*quoting Monroe Auto Equipment Co. v. Heckethorn Mfg. & Supply Co.*, 332 F. 2d 406, 412 (CA6 1964))).

Applicants' claimed subject matter relates to systems and methods for specifying and executing temporal order events. The loose specification of events gives users the maximum

ability to explore, create variations and make adjustments. The users need not specify exact literal times that an event or task is to be executed. Rather the users need only specify information that that they desire. The rest of the information is inferred by the system. To this end, independent claim 1 recites *a constraint component that receives loose temporal constraints associated with a plurality of events, wherein the loose temporal constraints specify information about execution of an event comprising a start time or a stop time and event execution relative to other events, a system information component that receives execution system information comprising available memory, cache coherency, data throughput and number of processors, and an order component that determines... a plurality of event orders in accordance with the loose temporal constraints* ***and selects an optimal event order based at least in part on execution system information, wherein the event order specifies the execution order of events***. While the examiner stated that the aforementioned limitation was anticipated by the cited art (Jerome *et al.* and Zweben *et al.*), applicants' representative would like to respectfully explain in detail distinctions between the cited art and the subject claims.

Jerome *et al.* provides method and system for creating and using a graphical task scheduler. A Graphical User Interface (GUI) allows a user to graphically build a real time flow task scheduler by providing a "click & drag" functional palette which contains graphical flow chart elements. The Examiner acknowledges that the primary reference, Jerome *et al.*, does not teach the claimed subject matter in its entirety with respect to *the fuzzy edges on at beginning of the bar indicate an unspecified start time and the fuzzy logic on at end of the bar indicates an unspecified end time and/or duration and hard bold edges on the bar specifies specific start and/or stop time* and provides a secondary reference, Zweben *et al.*, to compensate for the deficiencies of Jerome et al.. Zweben *et al.*, given by Examiner, relates to a method for modeling a complex enterprise, the operation of which is characterized by a large body of information. The method is used to develop a scheduling system for scheduling a complex activity and revising the schedule as necessary to accommodate changed circumstances; and this reference does not teach the claimed subject matter.

At page 20 of the Final Office Action, it is erroneously asserted that Zweben *et al.* substantially teaches that *the fuzzy edges on at beginning of the bar indicate an unspecified start time and the fuzzy logic on at end of the bar indicates an unspecified end time and/or duration*, with respect to dependent claim 41 and *hard bold edges on the bar specifies specific start and/or*

stop time, with respect to dependent claim 42. The reference (Zweben *et al.*) provides for resource history for a reusable resource pool. Initially, the quantity of the resource available in the resource pool is '5'. As illustrated by the bar above the resource history, task '1' begins at time '100' and requires '1' unit of the resource. The resource history decrements '1' unit from the pool and the resource pool has '4' units available from time '100' to time '200'. At time '200', Task '2' begins. Task '2' also requires '1' unit of the resource. Task 1 continues to take place, a total of '2' units are decremented from the resource pool. Consequently, the resource pool has only '3' units available from time '200' to time '300'. At time '300', both Task '1' and Task '2' conclude, releasing the resources that were being used. As a result, the pool is incremented 2 units so that 5 units are again available in the resource pool (*See*, Fig. 6). Hence Zweben *et al.* provides for resource history for a reusable resource pool by using bars. More particularly, Zweben *et al.* provides for only utilizing bars for indicating resources available for a reusable resource pool at any time. However Zweben *et al.* does not contemplate *the fuzzy edges on at beginning of the bar and at end of the bar indicate an unspecified start time and/or duration* and *hard bold edges on the bar specifies specific start and/or stop time.*

In view of at least the foregoing, it is clear that Jerome *et al.* and Zweben *et al.* fail to teach each and every aspect recited in independent claims 20, 21 and 40-42. Therefore, it is respectfully requested that this rejection of subject claims 20, 21 and 40-42 be withdrawn.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP543US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Bhavani S. Rayaprolu/
Bhavani S. Rayaprolu
Reg. No. 56,583

AMIN, TUROCY & CALVIN, LLP
57TH Floor, Key Tower
127 Public Square
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731